

APPLICATION  
FOR  
UNITED STATES LETTERS PATENT

TITLE: CENTRAL CONTROL OF MULTIPLE ADDRESS  
DOMAINS WITHIN A ROUTER

APPLICANT: RICHARD H. CRUMP AND JANET Y. DOONG

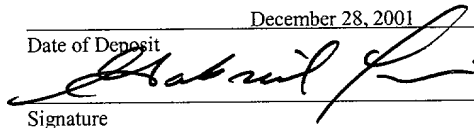
CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. EL584812132US

I hereby certify under 37 CFR §1.10 that this correspondence is being deposited with the United States Postal Service as Express Mail Post Office to Addressee with sufficient postage on the date indicated below and is addressed to the U.S. Patent and Trademark Office, P.O. Box 2327, Arlington, VA 22202.

December 28, 2001

Date of Deposit



Signature

Gabriel Lewis

Typed or Printed Name of Person Signing Certificate

## TECHNICAL FIELD

This invention relates to central control of multiple address domains within a router.

## BACKGROUND

[0001] Referring to FIG. 1, a router 10 is connected to networks 11, 12, 13, and 14 through interfaces 110, 120, 130, and 140, which are, for example, Ethernet cables. In a practical environment, router 10 can route packets through thousands of interfaces to thousands of networks.

[0002] Router 10 includes a routing table 15 for recording routing information. The routing information contains status and configurations of routes to which router 10 has access. The routes in routing table 15 can be determined according to a routing protocol, e.g., BGP (Border Gateway Protocol), OSPF (Open Shortest Path First), or RIP (Routing Information Protocol). BGP is an Internet protocol that allows routers across autonomous systems (e.g., company networks or university networks) to share routing information, thereby establishing efficient, loop-free routes. OSPF and RIP are routing protocols that allow routing information to be exchanged by routers within an autonomous system.

[0003] Router 10 also includes a routing table manager (RTM) 16 that performs central routing process and manages routing table 15. The central routing process comprises an IP (Internet Protocol) stack that runs router-executable processes, e.g., Unix processes. When a packet arrives at one of the interfaces 110, 120, 130, and 140, router 10 executes processes in the IP stack to perform specified functions according to the type of the packet. If the packet is a data packet, the IP stack selects a route from routing table 15, and forwards the packet according to the route. If the packet is a route update packet, as will be described below, the IP stack executes appropriate protocols to update routing table 15.

[0004] A route update packet is submitted to routing table manager 16 when a change occurs in a route accessible by router 10. A change in a route, for example, can be a configuration change in a remote network to which router 10 has access. Such a change can affect how router 10 routes incoming data packets to their respective destinations, and therefore is recorded in routing table 15.

[0006] The change in network 192.32.27.0 is submitted to routing table manager 16 through interfaces 120 and 130. If the change submitted by one interface, e.g., interface 120, arrives before the other interface, e.g., interface 130, routing table manager 16 will update routing table 15 according to the change submitted by interface 120. When another identical change subsequently arrives from interface 130, routing table manager 16 will treat the change as a new update. Routing table manager 16 will compare the later-arrived change with a corresponding entry in routing table 15, keep the one with better performance, and discard the other one.

**[0008]** Referring to FIG. 2, a router 20 includes two routing tables 151 and 152, each for one of the address domains, domain A and domain B. Router 20 is partitioned into two virtual routers 101 and 102, each functioning like a traditional router (e.g., router 10). Each of the virtual routers 101 and 102 maintains a stack, a routing table (151 or 152), and a routing manager (161 or 162). Information received from domain A propagates through an interface 110' or 120' to reach routing manager 161 and routing table 151, while information received from domain B propagates through an interface 130' or 140' to reach routing manager 162 and routing table 152. Each routing table records routes for one of the address domains. The two routing tables are independently updated, and information in one routing table is not shared with the other.

2

domain, running multiple IP stacks on one physical router 20 nevertheless limits the router's ability to scale.

### SUMMARY

[0010] In an aspect of the invention, a method for routing a packet comprises receiving the packet from one of a plurality of address domains through one of a plurality of interfaces; and determining one of a plurality of routing tables for the packet according to a mapping array. The mapping array includes pointers that associate the interfaces with the routing tables.

[0011] Embodiments of this aspect of the invention may include one or more of the following features.

[0012] The method further comprises executing a single IP stack to receive the packet and determine the one routing table. The mapping array associates interfaces connecting to the same address domain with the same routing table. After the one routing table is determined, the method further comprises forwarding the packet according to the one routing table if the packet is a data packet, or updating the one routing table if the packet is a route update packet. An example of the address domain includes a virtual private network

[0013] In another aspect of the invention, a router comprises: interfaces through which packets from address domains are received; and a domain manager that includes a mapping array for determining one of a plurality of routing tables for the received packets. The mapping array includes pointers that associate the interfaces with the routing tables.

[0014] Embodiments of this aspect of the invention may include one or more of the following features.

[0015] The domain manager executes a single IP stack to receive the packet and determine the one routing table. The mapping array associates interfaces connecting to the same address domain with the same routing table. The domain manager forwards the packet according to the determined one routing table if the packet is a data packet, or updates the determined one routing table if the packet is a route update packet. An example of the address domain includes a virtual private network.

[0016] In yet another aspect of the invention, a computer program product residing on a computer readable medium comprises instructions for causing the computer to: receive the packet from one of a plurality of address domains through one of a plurality of interfaces; and

determine one of a plurality of routing table for the packet according to a mapping array. The mapping array includes pointers that associate the interfaces with the routing tables.

[0017] Embodiments of this aspect of the invention may include one or more of the following features.

[0018] The computer program product further comprises instructions for causing the computer to execute a single IP stack to receive the packet and determine the one routing table. The mapping array associates interfaces connecting to the same address domain with the same routing table. After the one routing table is determined, the computer program product comprises instructions for causing the computer to forward the packet according to the one routing table if the packet is a data packet, or to update the one routing table if the packet is a route update packet. An example of the address domain includes a virtual private network.

[0019] The details of one or more embodiments of the invention are set forth in the accompanying drawings and the description below. Other features, objects, and advantages of the invention will be apparent from the description and drawings, and from the claims.

## DESCRIPTION OF DRAWINGS

[0020] FIG. 1 is a known router having a single routing table for a single address domain;

[0021] FIG. 2 is another known router partitioned as a plurality of virtual routers;

[0022] FIG. 3 is a router that uses a domain manager to dispatch packets to their respective routing tables;

[0023] FIG. 4 illustrates a mapping array used by the domain manager; and

[0024] FIG. 5 is a flow diagram that illustrates a process of the router for processing packets.

[0025] Like reference symbols in the various drawings indicate like elements.

## DETAILED DESCRIPTION

[0026] Referring to FIG. 3, router 30 is connected to networks 11, 12, 13 and 14 through interfaces 311, 312, 313, and 314, respectively. Networks 11 and 12 belong to domain A, and networks 13 and 14 belong to domain B. The address domains, for example, can be VPNs (Virtual Private Networks). In FIG. 3, each VPN is connected to router 30 through multiple interfaces. Each interface can be connected to a different business location of the VPN, or to the same location for the purpose of redundancy.

[0027] Router 30 includes a domain manager 36 that runs a single IP stack to perform central routing process. In contrast to router 10, which also includes only one routing table manager 16 running an IP stack, domain manager 36 manages access to more than one routing table. Routing tables 351 and 352 are domain-specific; that is, routing table 351 is used to handle packets propagating in domain A, while routing table 352 is used to handle packets propagating in domain B. Domain manager 36 can be implemented in software, hardware (e.g., ASICs (Application-Specific Integrate Circuits)), or a combination of both.

[0028] Domain manager 36 is responsible for dispatching packets coming from all interfaces 311, 312, 313, and 314 to appropriate routing tables (351 or 352). For each packet arriving at a given interface, domain manager 36 identifies an appropriate routing table by using a mapping array 39. Mapping array 39, being part of the domain manager's central control logic, includes pointers that associate a given interface to a routing table. Interfaces for the same address domain share the same routing table. For example, interfaces 311 and 312, which are both connected to domain A, submit incoming packets to the same routing table 351.

[0029] By using a separate routing table for each address domain, domain manager 36 assures privacy in routing information for multiple domains. If multiple interfaces that are connected to different address domains, e.g., interfaces 312 and 313, learn a route change in a network common to these address domains, e.g., domain A and domain B, the route change will be submitted to router 30 through both of the interfaces. Domain manager 36 will update each of routing tables 351 and 352 independently to reflect the change. The independent updates do not introduce scalability problems as exemplified by router 20, because router 30 only executes one IP stack to handle packets arriving from all address domains. The use of mapping array 39 allows domain manager 36 to efficiently associate interfaces to routing tables with only one IP stack.

[0030] Referring to FIG. 4, a router 40 includes a plurality of interfaces connected to a plurality of address domains. Router 40 includes a domain manager 46, which runs a single IP stack 42 and performs the same function as domain manager 36. Each interface (e.g., interface 410) of router 40 has an associated device driver 47, which manages incoming and outgoing bit streams. Device driver 47 assembles the incoming bit streams into packets in an IN\_queue 471 for domain manager 46 to fetch, and disassembles outgoing packets coming from the domain manager into bit streams in an OUT\_queue 472. To inform domain manager 46 of the interface

through which an assembled packet comes in, device driver 47 also inserts an interface number for its associated interface 410 in the packet before the packet is fetched by the domain manager. Based on the interface number, IP stack 42 locates a pointer in mapping array 49 that points to one of routing tables 45a, 45b, ..., 45n. Each of the routing tables 45a, 45b, ..., 45n is associated with interfaces that are connected to the same address domain.

[0031] Referring to FIG. 5, a flow diagram shows a process 50 for processing a packet at router 40. When bit streams arrive at any of the interfaces of router 40, e.g., interface 410, device driver 47 associated with that interface receives 51 the bit streams and assembles 52 them in IN\_queue 471 until the entire packet is completed. Device driver 47 also inserts 53 the interface number of interface 410 into the packet. Domain manager 46 executes IP stack 42 to read 54 packets from the IN\_queues of all the interfaces of router 40, including the packet in IN\_queue 471. The packets are dequeued after they are read. For the packet that arrives through interface 410, according to the interface number in the packet, IP stack 42 locates 55 a pointer in mapping array 49 that points to a routing table 45 corresponding to interface 410.

[0032] IP stack 42 also identifies 56 the packet as either a data packet or a control packet. If the packet is a data packet that needs to be forwarded to a destination network, domain manager 46 will look up 57 the corresponding routing table 45 for an appropriate route and an appropriate outgoing interface. The packet will be enqueued 58 in the OUT\_queue 472 of the outgoing interface, and the device driver 47 of that interface will disassemble the packet into bit streams and transmit 59 them on to a physical line that leads to the destination network.

[0033] If the packet is a control packet, the packet may contain a protocol update message or route update information that requires the corresponding routing table 45 to be updated. Domain manager 46 will forward 60 the packet up IP stack 42 to a routing protocol 43, e.g., BGP, as pre-determined by the address domain to which the packet belongs. According to the message or information specified in the control packet, routing protocol 43 updates 61 the corresponding routing table 45.

[0034] Accordingly, other embodiments are within the scope of the following claims.